MasterBase SW *3.x
MasterBase SW *4.x

CX System Messages & Halt Error Codes

Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER Hazards which could result in severe personal injury or death

WARNING Hazards which could result in personal injury

CAUTION Hazards which could result in equipment or property damage

NOTE Alerts user to pertinent facts and conditions

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Industrial Systems AB. ABB Industrial Systems AB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Industrial Systems AB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Industrial Systems AB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Industrial Systems AB's written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

Copyright © ABB Industrial Systems AB 1994, 1995, 1998

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1 Scope and Objectives 1-1
1.2 Structure of this document 1-1
1.3 Associated documents 1-2
1.4 Error handling principles in MasterBase SW 1-3
1.5 Reporting errors 1-3
1.6 Summary of technical changes 1-3

Chapter 2 - System halt error codes

2.1 General 2-1
2.2 Finding the system halt error code 2-1
2.3 System halt error code interpretation 2-2

Chapter 3 - System Message

3.1 General 3-1
3.2 Exceptions (mtype = 0..15) 3-3
3.3 MasterNet (mtype = 17 and 30) 3-9
3.3.1 MasterBus 300 or Generic Handler (mtype = 17, code = 1-9) 3-9
3.3.2 Symbol Translation (mtype = 17, code = 10) 3-15
3.3.3 Clock Synchronization (mtype = 17, code = 11) 3-16
3.3.4 Data Set, MVI-Data Set and Text Set Communication (mtype = 30) 3-17
3.3.5 EXCOM (mtype = 30) 3-21
3.3.6 MVI-Data Set with MVB API (mtype = 30) 3-22
3.4 Console communication (mtype = 18) 3-35
3.5 Basic I/O System errors (mtype = 20) 3-42
3.6 Operating System Kernel Utility (mtype = 29) 3-43

Chapter 4 - CPU Board Lamp Indicators

4.1 F 4-1
4.2 RUN 4-1
4.3 HLT 4-1
4.4 TO 4-1
4.5 DUAL 4-2
4.6 BAT 4-2
4.7 DCOK 4-2
4.8 INHIB 4-2
4.9 MIB 4-2
4.10 Seven segment displays 4-3

Chapter 1 Introduction

1.1 Scope and Objectives

This manual describes the general principles for error reporting in CX systems, and lists all system halt error codes and system messages that the MasterBase SW package can produce. The objective is to provide sufficient information to decide whether an error is caused by hardware failure, by mistakes in the use of MasterBase SW or by internal error in the software package.

For information concerning system messages produced by other software systems than MasterBase SW, please refer to the relevant user's manual.

For information on how to send system messages from application programs, please refer to CX System Services, User's Guide.

1.2 Structure of this document

The rest of this chapter introduces you to the error handling principles and error reporting principles that are used within the MasterBase SW. After that the following information can be found:

- Chapter 2 describes how system halt errors are reported and gives information that helps you to locate the error. This chapter also contains a complete list of halt error codes.
- Chapter 3 introduces you to the system message concept and gives a complete list of all system messages.
- Chapter 4 describes the meaning of the LED indicators on the CPU board and also how to interpret the 2-digit display that is present on PM510 and PM150 CPUs.

1.3 Associated documents

Documents that might be of special interest are presented very briefly below.

	Document Identity	Document Title
[1]	3BSE006397	CX Operating System General information This document shall be read to get an overview of the CX Operating system, especially with respect to its major components. You will also find the basic principles for task control and system supervision explained.
[2]	3BSE000253	CX System Services USER'S GUIDE In this manual you will find descriptions of every system service routine belonging to the MasterBase SW Operating system. The document is intended for programmers using the CX Operating system.
[3]	3BSE000255	CX System Generation USER'S GUIDE All parameters, macros etc. included in the System Generation Deck are detailed in this document. You need sometimes to compare the system message code with actual values entered into that deck to get the full picture of the error.
[4]	5761 0001-312	CX Console Communications USER'S GUIDE Introduces you to the Console Communications System of MasterBase SW and describes such things as command syntax, standard data types etc. You can read this manual to get further information on available documents within the Console Communications System area.
[5]	5761 0001-370	CX Data Base Accessor User's Guide This manual describes how to access the data base in programs using the data base accessor. In the manual each data base accessor procedure is described and different situations are shown in which the data base accessor is to be used.

There are also special documents for various parts of the system and you may sometimes need to consult those for additional information. This document does not contain a list of those document but it can be found in the release description for the relevant MasterBase SW release.

1.4 Error handling principles in MasterBase SW

Errors in an ABB Master software product are divided into two basic categories:

- System errors
- System components errors

A system error is an error that affects central parts of the system in such a way that further program execution is hazardous or impossible. When such errors occur the system is halted, and a system halt error code is saved in memory. In some cases additional diagnostic information is saved on the system stack.

A system component error is an error that affects only a limited part of the system, such as an input/output unit, a communication link or a user task. When such an error occurs the system component is separated from the system (put out-of-service, aborted), and the error is reported via a system message.

Certain errors can be handled by user tasks by means of exception handling. In such cases the application program itself decides whether to send a system message to report the error. More information on exception handling is found in CX System Services User's Guide.

1.5 Reporting errors

The MasterBase SW package is thoroughly tested, and is believed to be correct and reliable. However, in rare cases system halts or system messages may be caused by error in the system software. Such error codes and system messages are described in this manual with the text "Reason: internal error (please report to ABB Industrial Systems AB)".

If you receive an error code or a system message that indicates an error in MasterBase SW, please report it to ABB Industrial Systems AB by means of the System Problem Report (SPR) procedure. Be sure to include with the error report all available information on the problem.

SPR forms and a user's guide to the SPR procedure can be ordered from ABB Industrial Systems AB.

1.6 Summary of technical changes

This section summarizes changes done to the document since its last release.

MasterBase SW* 3.7/21 (Rev B of this document)
Description of system message 29:2 changed according to PMR SE 96 001412.

MasterBase SW* 4.1 (Rev B of this document) introduces support for MVI-Data Set with MVB API. Halt codes and description of system messages are added to this document.

Chapter 2 System halt error codes

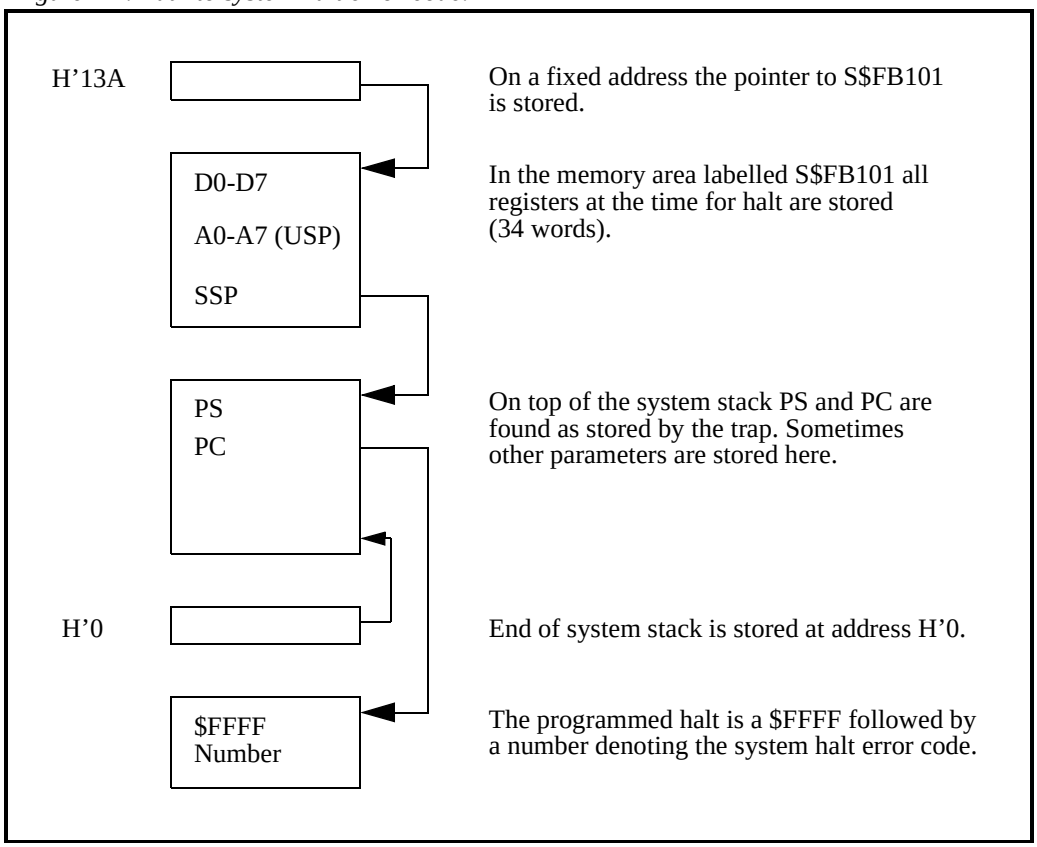
2.1 General

The programmed system halt is a “last resort” used by the operating system when no other error handling is possible. The intention is to completely inhibit all program activity, thereby stopping the error from propagating any further. As part of the programmed system halt the system’s state (i.e. CPU register contents etc.) is saved on defined locations in memory. By means of the “rudimentary console communication” the error can then be analyzed.

2.2 Finding the system halt error code

In all CX systems a pointer to certain system areas is stored at a fixed location. This pointer is used as a starting point in finding the cause of a programmed system halt. The procedure is illustrated in Figure 2-1 below.

Figure 2-1. Path to system halt error code.


H’13A
On a fixed address the pointer to S$FB101 is stored.
D0-D7
A0-A7 (USP)
SSP
In the memory area labelled S$FB101 all registers at the time for halt are stored (34 words).
PS
PC
On top of the system stack PS and PC are found as stored by the trap. Sometimes other parameters are stored here.
H’0
End of system stack is stored at address H’0.
$FFFF
Number
The programmed halt is a $FFFF followed by a number denoting the system halt error code.

2.3 System halt error code interpretation

The following is a listing of system halt error codes, with short explanations and suggested reasons:

code	meaning
H'0000	Debug trap in supervisory mode. Reason: probably memory or bus error.
H'0001	Trap in supervisory or kernel mode. Reason: probably memory or bus error.
H'0002	Trap H'E in supervisory or kernel mode. Reason: probably memory or bus error.
H'0003	Out of memory for task data areas. Reason: the RWM area is too small.
H'0004	Address error trap in system or kernel mode. Reason: probably memory or bus error. The system stack contains a CX standard address error trap information according to Figure 2-2.
H'0005	Bus time out error in system or kernel mode. Reason: probably memory error. The system stack contains a CX standard bus error trap information according to Figure 2-2.
H'0006	Reserved.
H'0007	Bus error: memory error (parity error). Reason: RWM error. The system stack contains a CX standard bus error trap information according to Figure 2-2.
H'0008	Reserved.
H'0009	Level 7 interrupt: stall alarm. Reason: If the CPU priority, as indicated by the processor status word (PS) stored on the system stack, is = 0 (task level), the reason is that the system is overloaded, or that a task contains an infinite loop, or that a task contains a programming error that makes it crash. If the CPU priority is > 0 (interrupt level) the reason may be a task level error as above, or an error in the interrupt system or on an interrupt generating unit, which causes an interrupt line to be stuck in an active state.
H'000A	RWM test error. Reason: error on read/write memory unit. The address of the faulty location is saved as register A0.
H'000B	PROM checksum error. Reason: error on PROM unit. The address of the Kilobyte that contains the faulty location is saved as register A0.
H'000C	Autotest error: instruction test failed. Reason: error on CPU card.

code	meaning
H’000D	Autotest error: interval timer test failed. Reason: error on CPU card.
H’000E	Autotest error: interrupt controller test failed. Reason: error on CPU card.
H’000F	Autotest error: bus error test failed. Reason: error on CPU card.
H’0010	Level 7 interrupt: power failure. Reason: power up after power fail in a system that has no power fail handling, i.e. neither warm start up nor power fail restart.
H’0011	Reserved.
H’0012	Reserved.
H’0013	Reserved.
H’0014	Reserved.
H’0015	Reserved.
H’0016	Scan task exit supervision halt. Reason: a task has done exit which is specified to halt the system at exit.
H’0017	Wrong CPU type in the system. Reason: The generated system does not match with the used CPU.
H’0018	Reserved.
H’0019	Reserved.
H’001A	Reserved.
H’001B	Reserved.
H’001C	Reserved.
H’001D	Reserved.
H’001E	Reserved.
H’001F	Reserved.
H’0020	Reserved.
H’0021	Reserved.
H’0022	Reserved.
H’0023	User defined or operator requested system halt.
H’0024	Backup halt due to PRIMARY failure.
H’0025	Reserved.
H’0026	Reserved.
H’0027	Reserved.

code	meaning
H’0028	Reserved.
H’0029	Reserved.
H’002A	Reserved.
H’002B	Reserved.
H’002C	Unimplemented trap in backup environment.
H’002D	Reserved.
H’002E	Reserved.
H’002F	Reserved.
H’0030	Reserved.
H’0031	Reserved.
H’0032	PBC access logging errors. D7=1 Mismatch between accessed and FIFO received data from PBC. D7=2 Mismatch between accessed and logged address strobes. D7=3 Strobe signals not logged in log buffer. D7=4 Illegal size code and access address combination.
H’0033	Previous PRIMARY halted on manual change-over. (Not a fault indication.)
H’0034	Reserved.
H’0035	Reserved.
H’0036	Programming error.
H’0037	No compressed data found, to expand, at Cold Start.
H’0038	Backup Prom is not equal to Primary Prom. (Halt code from Backup processor only.)
H’0039	Reserved.
H’003A	Reserved.
H’003B	Reserved.
H’003C	Reserved.
H’003D	Reserved.
H’003E	Reserved.
H’003F	Reserved.
H’0040	Redundant CPU. Initializing error.
H’0041	Redundant CPU. Testing error.
H’0042	Redundant CPU. Threading error (for BACKUP).
H’0043	Redundant CPU. RCU HW error.
H’0044	Redundant CPU. General Purpose Communication Channel error.

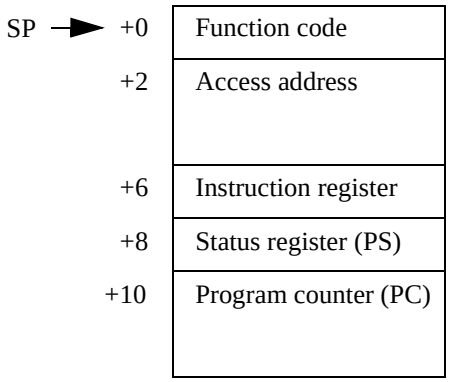
code	meaning
H’0045	Redundant CPU. Unknown interrupt reason.
H’0046	Redundant CPU. Floating Point error during take-over.
H’0047	Redundant CPU. MOVEM instruction to/from I/O during re-execution.
H’0048	Redundant CPU. Status channel interrupt not pending.
H’0049	Redundant CPU. Error during role/mode selection. D7=1 Unknown negotiation step taken by other board D7=2 Error at attaching interval timer exception routine D7=3 Error at starting the interval timer D7=5 Error saving the chosen role in the non-volatile memory object D7=8 Error resetting the suspected bit in the non-volatile memory object
H’004A	Reserved.
H’004B	Reserved.
H’004C	Reserved.
H’004D	Reserved.
H’004E	Reserved.
H’004F	Reserved.
H’0050	Common System Start-up. System bus error.
H’0051	Common System Start-up. Display object init error.
H’0052	Common System Start-up. CPU object init error.
H’0053	Common System Start-up. Bus time-out timer object init error.
H’0054	Common System Start-up. Interval timer object init error.
H’0055	Common System Start-up. Stall timer object init error.
H’0056	Common System Start-up. Start Mode Selector object init error.
H’0057	Common System Start-up. Memory object init error.
H’0058	Reserved.
H’0059	Common System Start-up. Boot PROM checksum error.
H’005A	Common System Start-up. Boot RWM test error.
H’005B	Reserved.
H’005C	Reserved.
H’005D	Reserved.
H’005E	Reserved.
H’005F	Reserved.

System halt codes H’0060 to H’007F are reserved for MOD 300.

System halt codes H’0080 to H’00FE are reserved for MasterBase.

code	meaning
H’00FF	Not programmed system halt.

Figure 2-2. CX standard address/bus error stack frame



At address error and bus error the address/bus error stack frame is converted to a CPU independent (CX standard) stack frame. This figure shows the layout of the CX standard address-error/bus-error stack frame.

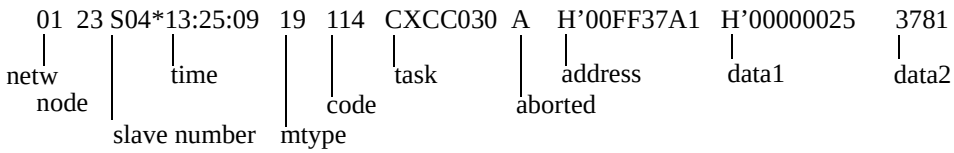
Chapter 3 System Message

3.1 General

System messages are used to report errors and similar events that have been discovered by the system. They give information on the type of error, where the error occurred and in many cases also additional diagnostic information to indicate what caused the error.

A system message can be printed out by the system that generated it, or it can be transported by means of computer network communication and printed out by another system, or it can be displayed on a service aid. Therefore each system message contains the network and node identities of the system in which it was generated.

System message printouts have the following appearance:



Network and node number is not printed if the message comes from the node where the output task resides. Slave number is not printed if the message comes from the master. The abort indication, address, data1 and data2 can all be missing; in that case the corresponding part is blanked.

Mtype specifies message type. The following mtypes are used:

mtype	category	
0	Parameter error on system service call	Exceptions
1	Address and illegal trap error	
2	Insufficient resources	
3	Bus error	
4	Power fail	
5	Task killed	
6	Code conversion error	
7	Time out after SENDW	
8	Data base error	
9	I/O error	
10	Abort	
11	Stack overflow	
12	Floating point error	
13	Spare	
14	Spare	
15	Spare	
16	Spare	

Mtypes 17 - 255 are used as follows (for explanations, please refer to the relevant user's manual):

17	Computer network communication
18	Console communication
19	Reserved
20	Basic I/O System
21..22	PC process control package
23..26	MMC man machine communication package
27	DCM service aids
28	Process communication
29	Operating System Kernel Utility
30	MasterBus, DSC, EXCOM, Clock Synch.
31	Spare (reserved for basic software)
32..63	ABB Master standard product specific
64..255	Application specific

3.2 Exceptions (mtype = 0..15)

Exceptions are errors detected by the system software while a task is executing, such as parameter errors in system service calls, bus errors, insufficient resources etc. Normally exceptions are handled by the system software, but a user task may to a certain extent take care of exceptions itself by requesting “exception handling”, see CX System Services User’s Guide, Reference [2].

Exceptions handled by the system software are reported with system messages with mtype 0..15. The following error codes are used in the “code” field:

code	meaning
0	On system start-up after power fail: the system was started with a warm start-up condition. Addr = - Data1 = - Data2 = -
2	On system start-up after power fail: the system was started with a power fail restart condition. Addr = - Data1 = duration of power failure in ms (decimal representation) Data2 = -
3	Port number error: the specified task has referred to a port that does not exist. Addr = location of error (absolute address) Data1 = - Data2 = -
4	Signal priority error: the specified task has used an illegal signal priority. Addr = location of error (absolute address) Data1 = - Data2 = -
5	No SQB: no Signal Queue Block was available (i.e. the channel was full) when the specified task tried to send a signal. Addr = location of error (absolute address) Data1 = - Data2 = -
6	No TQB: no Timer Queue Block was available when the specified task tried to send a signal with SENDT. Addr = location of error (absolute address) Data1 = - Data2 = -
7	Time-out on SENDW: no receiving task has acknowledged the signal sent with SENDW by the specified task. Addr = location of error (absolute address) Data1 = - Data2 = -

code	meaning
9	Parameter error in SENDR or SENDW call. Error when response part of signal was accessed (signal data in task data area). Addr = location of error (absolute address) Data1 = - Data2 = -
10	Error in time parameter: a time parameter used by the specified task is not a legal time interval or point-of-time. Addr = location of error (absolute address) Data1 = - Data2 = -
11	Response already given: the specified task has sent a response on a received signal, but a response has already been sent by another receiver task on the same channel. Addr = location of error (absolute address) Data1 = - Data2 = -
12	Inport not in expected state: the task does not handle its inport(s) in the correct way (e.g. an ENDREC was executed when no signal was received) Addr = location of error (absolute address) Data1 = - Data2 = -
13	Operation not allowed for multiple inports: the task has used a system service that is not defined for multiple inports. Addr = location of error (absolute address) Data1 = - Data2 = -
14	Stack overflow. Addr = - Data1 = old stack pointer value Data2 = -
15	Error in BIOS system service call parameter. Addr = location of error (absolute address) Data1 = - Data2 = -
16	No IQB with correct state available. Addr = location of error (absolute address) Data1 = - Data2 = -
17	Illegal device reserve operation: the task already has a device reserved. Addr = location of error (absolute address) Data1 = - Data2 = -

code	meaning
18	Error reported by handler on BIOS operation in wait mode. Addr = location of error (absolute address) Data1 = BIOS operation code (hexadecimal representation) Data2 = most sign word: LU number (hexadecimal representation) least sign word: IQB status (hexadecimal representation)
19	Aborted: the task has aborted itself. Addr = location of ABORT call (absolute address) Data1 = abort code given by the task in ABORT call (decimal representation) Data2 = -
20	Bus error. Addr = address within error generating instruction Data1 = process status word Data2 = the faulty effective address
21	Killed: the task has been killed.
22	Error trap, e.g. address error (odd addr), not implemented instruction error etc. Addr = location in trapping program (abs addr) Data1 = process status word Data2 = trap vector address / odd effective address
23	Illegal trap. Addr = location in trapping program (abs addr) Data1 = processor status word Data2 = trap vector address
24	Pascal runtime error. Addr = location of error (absolute address) Data1 = Pascal runtime error code (decimal representation) In addition to the pascal runtime error codes given in Motorola Pascal user's manual, the following codes are defined by CX: 1015 = Run-time routine is missing: the specified task calls a run-time routine that was not included when the system was generated. 1071 = Local or indexed file not supported: an attempt has been made to use a local or an indexed file. This error will occur when the file list in the PROGRAM statement has been omitted in a program that uses Pascal files. 1072 = No logical unit found: there is no logical unit defined in the task's parameter area (see MasterBase SW, System generation user's manual) 1073 = I/O error open (reset, rewrite). This error is caused by a hardware error or by an assignment error. 1074 = I/O error read. This error is caused by a hardware error or by an assignment error. 1075 = I/O error write. This error is caused by a hardware error or by an assignment error. Data2 = Statement number (if K-option)

code	meaning
25	Access Control Block error: error in use of an Access Control Block (ACB). Addr = - Data1 = see CX Data Base Accessor User's Guide. Data2 = see CX Data Base Accessor User's Guide.
26	Illegal concept: the concept is not defined in the system. Addr = - Data1 = see CX Data Base Accessor User's Guide. Data2 = see CX Data Base Accessor User's Guide.
27	Illegal logical file: the file is not defined in the system. Addr = - Data1 = see CX Data Base Accessor User's Guide. Data2 = see CX Data Base Accessor User's Guide.
28	Illegal logical record: the record number is greater than what is defined as maximum for the logical file. Addr = - Data1 = see CX Data Base Accessor User's Guide. Data2 = see CX Data Base Accessor User's Guide.
29	Illegal property number: the property is not defined. Addr = - Data1 = see CX Data Base Accessor User's Guide. Data2 = see CX Data Base Accessor User's Guide.
30	Illegal member number: the member number is outside the range that is defined for the repeating group. Addr = - Data1 = see CX Data Base Accessor User's Guide. Data2 = see CX Data Base Accessor User's Guide.
31	Illegal variant: the variant is not defined. Addr = - Data1 = see CX Data Base Accessor User's Guide. Data2 = see CX Data Base Accessor User's Guide.
32	Illegal physical file: the physical file is not defined, or the physical file type is not correct. Addr = - Data1 = see CX Data Base Accessor User's Guide. Data2 = see CX Data Base Accessor User's Guide.
33	Illegal group type. Addr = - Data1 = see CX Data Base Accessor User's Guide. Data2 = see CX Data Base Accessor User's Guide.

code	meaning
34	Error in action routine: an error has occurred in an action routine associated with an item accessed by the task. Addr = - Data1 = see CX Data Base Accessor User's Guide. Data2 = see CX Data Base Accessor User's Guide.
35	Function not implemented in the system: the system does not support the function referred to (system generation error). Addr = - Data1 = see CX Data Base Accessor User's Guide. Data2 = see CX Data Base Accessor User's Guide.
36	Illegal floating point operation. Addr = location of error Data1 = processor status word Data2 = Pascal runtime error code: 2001: illegal operation 2002: overflow 2008: division by zero 2020: integer overflow
37	Overflow in floating point operation. Addr = location of error Data1 = processor status word Data2 = Pascal runtime error code: 2001: illegal operation 2002: overflow 2008: division by zero 2020: integer overflow
38	Underflow in floating point operation (never generated).
39	Division by zero in floating point operation. Addr = location of error Data1 = processor status word Data2 = Pascal runtime error code: 2001: illegal operation 2002: overflow 2008: division by zero 2020: integer overflow
40	Inexact in floating point operation (never generated).
41	Integer overflow in floating point operation. Addr = location of error Data1 = processor status word Data2 = Pascal runtime error code: 2001: illegal operation 2002: overflow 2008: division by zero 2020: integer overflow

code	meaning
42	Illegal network or node number in dsx.header.
43	Not used.
44	Illegal message length in dsx.header.
45	Illegal destination channel identity in dsx.header.
46	System overloaded, i.e. time-out for the idle task (cf. low level stall).
47	Illegal release of a semaphore: If the semaphore is used for protection of common resources (binary semaphore), only that task who has taken the semaphore is allowed to release it. Addr = the program counter value when abort was issued. Data1 = pointer to the semaphore that the task tried to release but did not own. Data2 = -

3.3 MasterNet (mtype = 17 and 30)

3.3.1 MasterBus 300 or Generic Handler (mtype = 17, code = 1-9)

All MasterNet system messages in MP 200, MV 800 or MG 230 stations use mtype 17 (MasterBus 300).

code — meaning

1 — Configuration error.
Data1 = error code:

- 1 = internal error: illegal length of control block array
- 2 = illegal block size in buffer pool
  Data2 = pool identifier
- 3 = Reserved
- 4 = illegal buffer class parameters
  Data2 = pool identifier
- 5 = illegal number of classes
  Data2 = pool identifier
- 6 = internal error: number of pools inconsistent
- 7 = insufficient number of buffer control blocks
  Data2 = pool identifier
- 8 = insufficient number of allocation wait request blocks
  Data2 = pool identifier
- 9 = insufficient number of buffer blocks
  Data2 = pool identifier
- 10 = insufficient number of data link user control blocks
- 12 = data link can't use buffer pool the data link handler will abort
  Data2 = internal error code (Hex)
  - 0F = dlsdu size for data link is too large or initialization of buffer pool has failed
  - 30 = the pool identifier for data link is illegal
- 14 = error in network connection
  Data2 = H'RRNENWNO
  - RR = reserved
  - NE = network number strapped on CS513.
  - NW= source network number in received route message
  - NO = source node number in received route message

  CS513 is connected to wrong transceiver or error in network number address on CS513
- 17 = data base is full with path descriptions
  Data2 = data base record number
- 18 = network path array is full
  Data2 = data base record number

code	meaning
1 (cont.)	Configuration error. Data1 = error code: 19 = a path conflict between data base and network path array is detected Data2 = data base record number 21 = duplicate station address on MasterBus 300 Data2 = node number Change node number 22 = more than one node number in the station Data2 = subnet and node number Change node number 23 = station connected to more than one Data2 = subnet and node number Change subnet number if the station is to be connected in a Control Network. Change node number if the station is to be connected in the Plant Network. 24 = slave processor board not compatible with slave handler software in host processor. Data2 = slave number and subnet number 25 = duplicate network number in station Data2 = subnet and node number Change network number 27 = Different offsets in S$MYDLSAP entries Data2 = current offset/first offset 50 = the slave number is out of range. Data2 = H'LRSLCPSP LR = Logical record in data base. SL = Slave number. CP = Carrier position number. SP = Submodule position on carrier. 51 = the MVI carrier or submodule position defined in database is out of range. Data2 = the same as for "error 50" 52 = the MVI carrier and submodule position defined in data base is already used by another submodule. Data2 = the same as for "error 50" 53 = the slave number is already used by another submodule. Data2 = the same as for "error 50" 54 = the LAN channel is already used by another LAN submodule. Data2 = the same as for "error 50" but SL = 9X with X = channel number.

code	meaning
2	Resource error. Data1 = type of resource 9 = insufficient number of buffer blocks Data2 = pool identifier 15 = insufficient number of path connection control blocks 16 = insufficient number of network connection or path connection control blocks 18 = too many stations connected to MasterBus 300 Data2 = subnet number 19 = error in allocation of memory at system start up Data2 = 0 host processor >0 slave processor number 21 = No place in S$NETNOD for new netw/node numbers Data2 = netw/node 22 = No place in DCB for netw/node numbers Data2 = netw/node 23 = No free block in S$SLCHAN Data2 = netw/node 24 = No free block in the host S$DISTANCE table Data2 = slave number 25 = No free block in the slave's S$CROSSREF table Data2 = DLSAP index 26 = No free block in the host S$EXTNOD table. Error in allocation of memory to MAC receive ring.

code meaning

3 Disconnect
Data1 = type of connection
11 = data link disconnected
Addr = disconnect reason
HEX 5200 - 5500 reserved
Data2 = disconnect reason
1 = internal error;
please report to ABB Industrial Systems AB.
2 = MAC transceiver heart beat failure.
Check physical connection.
3 = DMA error. Change communication board.

13 = network path disconnected
Data2 = H'RRAANENO
RR = disconnect reason
AA = alternative number for path
00 = all paths to destination
NE = network number
NO = node number

Disconnect reasons (Hex):
00 = destination node unreachable

7 Re-connect
Data1 = type of connection
13 = network path re-connected
Data2 = H'0000NENO
NE = subnet number
NO = node number

8 Hardware error
Data1 = communication board
01 = DSCS 140/DSCS 145
Data2 = H'SLTTNENO
SL = slave number
TT = type of error
01 = LANCE failed to initialize;
SL and NE = 00
02 = LANCE BUFF error in transmit
MAC packet. Hardware error or
electrical interference on coax or
drop cable.
SL and NE = 00
03 = If START switch is CLEAR then
change DSCS 140 board
NE = network number
NO = node number

code	meaning
8 (cont.)	Hardware error Data1 = communication board 02 = IEEE 802.3 MAC error Data2 = H'CCSSEEMM CC = Carrier position in station rack SS = LAN board subposition on carrier EE = type of error 03 = Checksum error in received frame. 04 = Board missing or not accessible. 0A = Late collision while transmitting frame. 0B = Loss of Carrier 0D = Transmit retry error. CCSS = Time Domain Reflectometry. 10*CCSS gives distance in meter to suspected cable fault. 44 = Controller failed to initialize. 48 = Babble error. 49 = Transceiver Heart beat error. The transceiver is not properly connected or the transceiver SQE test is not working. 4A = Memory error 4D = Controller is restarted MM= MAC channel number 03 = SC520/SC530 Data2 = H'CC00EE00 CC = Board position in station rack EE = Type of error 01 = failed to initialize 03 = board has halted 04 = board missing or not accessible 04 = CI532/CI535 Data2 = H'CCSSEE00 CC = Carrier position in station rack SS = Board subposition on carrier EE = Type of error 01 = failed to initialize 03 = board has halted 04 = board missing or not accessible

code	meaning
8 (cont.)	Hardware error Data1 = communication board 05 = CI531 Data2 = H'CCSSEE00 CC = Carrier position in station rack SS = Board subposition on carrier EE = Type of error 01=failed to initialize 02 = board has halted 03 = board missing or not accessible
9	Hardware operable after failure Data1 = communication board 01 = DSCS 140 or DSCS 145 Data2 = H'SLTTNENO SL = slave number TT = 00 NE = network number NO = node number 02 = IEEE 802.3 MAC error Data2 = H'CCSS0000 CC = Board position in station rack SS = Board subposition on carrier 03 = SC520/SC530 Data2 = H'CC000000 CC = Board position in station rack 04 = CI532/CI535 Data2 = H'CCSS0000 CC = Carrier position in station rack SS = Board subposition on carrier 05 = CI531 Data2 = H'CCSS0000 CC = Carrier position in station rack SS = Board subposition on carrier

3.3.2 Symbol Translation (mtype = 17, code = 10)

System messages sent by the Symbol Translation (MasterNet Directory Service) use mtype 17.

code	meaning
10	Duplicate symbol name
	Data1 = node number, this is a node which has a symbol with the same name as the one previously solved in data base.
	Data2 = H'FFFFRRRR FFFF = Logical file in Data Base RRRR = Logical record in Data Base

3.3.3 Clock Synchronization (mtype = 17, code = 11)

All system messages sent by the Clock Synchronization system use mtype 17 and code 11. The content of address field for all of them is a message identification number rather than absolute address indicating where the error is detected.

code	meaning
11	Clock synchronization message Data1 = error code: 1 = Clock Synch channels not found. Data2 = 0 2 = Failed to select Clock Master node. No neighbour with correct parameters. Data2 = 0 3 = There might be more than one backup node. Set time again! Data2 = backup node 4 = More than one Clock Master in network. Set time again! Data2 = 0 5 = High precision time accuracy. (If Clock Synch log flag set). Data2 = current time error in 0.1 ms Addr = AAAABBBB AAAA = Max occurred error in 0.1 ms during period BBBB = Mean error during period

3.3.4 Data Set, MVI-Data Set and Text Set Communication (mtype = 30)

System messages from the Data Set Communication, MVI - Data Set Communication and Text Set Communication are equal except from the task name in the message.

Definitions used in this section:

APT request	One shot request for a single DSP from an application task.
DR	Data Record
DSC	Data Set Communication.
DSD	Data Set Descriptor. Identifies the item in the data base whose values are to be collected or up- dated by DSC.
DSP	Data Set Package. A collection of data taken from the DR-file.
MVI_DSC	Multi Vendor Interface - Data Set Communication
MVI_DSD	MVI - Data Set Descriptor. Identifies the item in the data base whose values are to be collected or updated by MVI_DSC.
TEXT_DATA	Text data file
TSC	Text Set Communication
TSD	Text Set Descriptor. Identifies the item in the data base whose values are to be collected or up- dated by TSC.
TSP	Text Set Package. Allocation of data taken from the TEXT_DATA file.

System messages from data set, MVI-data set and text set communication have the following format:

<mtype> <code> <taskname> <address> <data1> <data>

<mtype> = 30

<code> = error code as described below.

<taskname> = task name
CXAA000 = Message from Data Set Communication
CXAM000 = Message from MVI - Data Set Communication
CXAT000 = Message from Text Set Communication

<address> = message identification number in hexadecimal representation (not the address where the error is located).

<data1> = detailed information about the error.

<data2> = parameter, in hexadecimal representation, which locates the error (e.g. data base index). Its exact format depends on the particular message.

code	meaning
19	Abort Addr = Internal address when aborted Data1 = 61000 Holes in DSD, MVI_DSD or TSD file. 61001 Data references does not exist. 61002 Can not get buffer space (system data area too small). 61700 No records in concept (rec. no 1 missing) DSD, DR, MVI_DSD or TSD. Data2 = -
20	Configuration error. This message can not be generated from Text Set Communication. Addr = H'EDE4 (= 60900) Data base contains receive type DSD or MVI_DSD but configuration specifies no receive type. DSD or MVI_DSD has been blocked. Data1 = 2 (operation not allowed) Data2 = DSD or MVI_DSD number Addr = H'EF10 (= 61200) APT request received but not specified in system. Data1 = 2 (operation not allowed) Data2 = Not used Addr = H'EF74 (= 61300) Cyclic scanning not included in the system but data base holds a DSD or MVI_DSD of source type. DSD or MVI_DSD has been blocked. Data1 = 2 (operation not allowed) Data2 = DSD or MVI_DSD number Addr = H'EFD8 (= 61400) Receiving DSP not allowed in the system but DSP received. Data1 = 2 (operation not allowed) Data2 = H'XXYYZZ00 XX = Source network YY = Source node ZZ = DSD or MVI_DSD number
21	Lack of resources Addr = H'EEAC (= 61100) Channel to handler is full. Data1 = 1 (overflow in channel) Data2 = H'XXYY0000 XX = Destination network YY = Destination node

code	meaning
23	Database error

Addr = H'EAC5 (= 60101)
APT request refers to DSD or MVI_DSD with illegal scan factor.
Data1 = 4 (bad or inconsistence structure)
Data2 = DSD or MVI_DSD number

Addr = H'EB28 (= 60200)
DSD, MVI_DSD or TSD has no data references. DSD or MVI_DSD has been blocked.
Data1 = 4 (bad or inconsistence structure)
Data2 = DSD, MVI_DSD or TSD number

Addr = H'EBF0 (= 60400)
Cyclic scan - given DSD or MVI_DSD has illegal scan factor. DSD or MVI_DSD has been blocked.
Data1 = 4 (bad or inconsistence structure)
Data2 = DSD or MVI_DSD number

Addr = H'ED1D (= 60701)
Illegal scan factor for receive type DSD or MVI_DSD. DSD or MVI_DSD has been blocked.
Data1 = 4 (bad or inconsistence structure)
Data2 = DSD or MVI_DSD number

Addr = H'ED80 (= 60800)
Amount of data references in received DSP or TSP does not agree with referenced DSD, MVI_DSD or TSD. DSD or MVI_DSD has been blocked.
Data1 = 4 (bad or inconsistence structure)
Data2 = H'XXXXYYYY
XXXX = Number of data references in DSD, MVI_DSD or TSD.
YYYY = DSD, MVI_DSD or TSD number

Addr = H'EE49 (= 61001)
DSD, MVI_DSD or TSD data reference does not exist. Applies even to non-enabled DSDs, MVI_DSDs and TSDs.
Data1 = 4 (bad or inconsistence structure)
Data2 = H'XXXXYYYY
XXXX = Reference in DSD, MVI_DSD or TSD.
YYYY = DSD, MVI_DSD or TSD number

code	meaning
23 (cont.)	Database error

Addr= H'EEAD (= 61101)
Destination in DSD, MVI_DSD or TSD currently unreachable. Can occur as a result of either Text request, APT request or cyclic scan.
Data1 = 4 (bad or inconsistence structure)
Data2 = DSD, MVI_DSD or TSD number

Addr = H'F1CC (= 61900)
Addressed DSD, MVI_DSD or TSD not found. Occurs during Text request or APT request (source type) or receive DSP, MVI_DSP or TSP (receive type).
Data1 = 3 (structure not found)
Data2 = H'XXYYZZWW
XX = Destination network
YY = Destination node
ZZ = DSD, MVI_DSD or TSD number
WW = 1 -> source type
0 -> receive type

3.3.5 EXCOM (mtype = 30)

All system messages sent by the EXCOM system use mtype 30. The content of address field for all of them is a message identification number rather than absolute address indicating where the error is detected.

code	meaning
21	Lack of resources when calling CXSENDQ. The message will only be sent the first time the error is detected. Addr = H'65000 Data1 = destination network Data2 = destination node

3.3.6 MVI-Data Set with MVB API (mtype = 30)

Introduced in MasterBase SW*4.1. All system messages sent by the MVI-Data Set with MVB API system use mtype 30.

Definitions used in this section:

API	Application Interface
APT request	One shot request for a single DSP from an application task.
CXAX	Common name for the tasks. Multi Vendor Interface - Data Set Communication with Multi Vendor Block Application Interface
DR	Data Record
DSP	Data Set Package. A collection of data taken from/to the DR-file.
MVB	MVI Vendor Block. Identifies the item in the data base whose values are to be collected or updated by MVI-Data Set task.
MVI_MODULE	Multi Vendor Interface Module.
MVICHAN	Multi Vendor Interface Channel
MVINODE	Multi Vendor Interface Node

System messages from MVI-Data Set with MVB API have the following format:

<mtype> <code> <taskname> <address> <data1> <data>

<mtype> = 30

<code> = error code as described below.

<taskname> = task name
CXAX000 = Message from the MVI-Data Set distribution task
CXAX01xy= Message from the MVI-Data Set tasks,
where x is submodule number
(1-2 for AC410 and 1-5 for AC450).
where y is channel number (1-2)

<address> = message identification number in hexadecimal representation (not the address where the error is located).

<data1> = detailed information about the error.

<data2> = parameter, in hexadecimal representation, which locates the error (e.g. data base index). Its exact format depends on the particular message.

code	meaning
19	**Abort**

Addr = Internal address when aborted

Data1 = 60340 Memory allocation error (wRspTblP)
- 60401 Record no 1 missing (MVI_MODULE, MVICHAN, MVINODE, DR, MVB)
- 60405 MVI_MODULE for this task instance is missing or not correctly initialized.
- 60410 MVI_MODULE for this task instance is missing or not correctly initialized.
- 60420 Memory allocation error (destRecP)
- 60421 Memory allocation error (cyclRecP)
- 60422 Unknown task number
- 60423 MVI_MODULE for this task instance is missing
- 60424 Holes in MVB file
- 60425 Data reference does not exist
- 60426 Holes in MVINODE file

Data2 = -

20 **Configuration error.**

Addr = H'EB46 (=60230)
Attempt to send signal to unreachable node. MVB is marked with error.

Data1 = 3 (CzcNoHit - structure not found)

Data2 = H'XXXXYYYY
- XXXX = destination node number
- YYYY = MVB number

21 **Lack of resources**

Addr = H'EAC5 (=60101)
'Wait send list' is full. MVB not inserted into list. Occurs during APT request.

Data1 = -

Data2 = H'XXYYZZ00
- XX = Network number
- YY = Node number
- ZZ = MVB Ident number

code	meaning
21 (cont.)	Lack of resources

Addr = H'EBAA (=60330)
Sending protocol information to MVI module wasn´t successful.

Data1 = 1 (CzcFullChan - overflow on channel)

Data2 = H'XXYY00ZZ
XX = destination network
YY = destination node
ZZ = MVICHANs network number

Addr = H'ECB8 (=60600)
Exception handling. Overflow on channel. Node number is negative.

Data1 = 1 (CzcFullChan - overflow on channel)

Data2 = H'XXYY0000
XX = Destination network
YY = Destination node

Addr = H'ECBA (=60602)
Exception handling. Overflow on channel. Node number is positive.

Data1 = 1 (CzcFullChan - overflow on channel)

Data2 = H'XXYY0000
XX = Destination network
YY = Destination node

Addr = H'1110D (=69901)
Channel to handler is full. Forwarding the APT Request from CXAX000 to CXAX01xy wasn't successful.

Data1 = 1 (CzcFullChan - overflow on channel)

Data2 = H'XXYYZZWW
XX = destination network
YY = destination node
ZZ = MVI_MODULE instance number
WW = port number

code	meaning
23	**Database error**

Addr = H'EAC4 (=60100)
APT request refers to MVB with illegal scan factor.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

Addr = H'EACF (=60111)
A 'Down Stream Response' signal has been received. The Command Code in the Response is of type Read, but the Source bit is set in the MVB.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

Addr = H'EAE2 (=60130)
Cyclic scan - given MVB has illegal scan factor. MVB is set to be blocked.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

Addr = H'EAF8 (=60152, displayed only if DS Log-flag is on)
An 'Up Stream Command' signal has been received. The Command Code in the signal is of type Read, but the Source bit is not set in the MVB.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

Addr = H'EAF9 (=60153, displayed only if DS Log-flag is on)
An 'Up Stream Command' signal has been received. The Command Code in the signal is of type Write, but the Source bit is set in the MVB.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

code	meaning
23 (cont.)	Database error

Addr = H'EB29 (=60201)
Unknown Commad Code in MVB. Occurs during building an Data Set Package requested from APT or cyclic sending.

Data1 = 6 (CzcCmdCode - error in command code)

Data2 = H'XXYYZZZZ
XX = Network number
YY = Node number
ZZZZ = Command Code

Addr = H'EB2A (=60202)
Source bit is set in the MVB. Wrong value, according to Command Code. Occurs during building an Data Set Package requested from APT or cyclic sending.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = H'XXYYZZZZ
XX = Network number
YY = Node number
ZZZZ = Command Code

Addr = H'EB2B (=60203)
Source bit is not set in the MVB. Wrong value, according to Command Code. Occurs during building an Data Set Package requested from APT or cyclic sending.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = H'XXYYZZZZ
XX = Network number
YY = Node number
ZZZZ = Command Code

Addr = H'EB2C (=60204)
Illegal register address in MVB. DatRef and BitRef are included. Command code is of type INT. Occurs during building an Data Set Package requested from APT or cyclic sending.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

code	meaning
23 (cont.)	Database error

Addr = H'EB2D (=60205)
Illegal register address in MVB. DatRef and BitRef are included. Command code is of type INTL, REAL or MIXL. Occurs during building an Data Set Package requested from APT or cyclic sending.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

Addr = H'EB2E (=60206)
Illegal register address in MVB. DatRef and BitRef are included. Command code is of type BOOL. Occurs during building an Data Set Package requested from APT or cyclic sending.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

Addr = H'EB2F (=60207)
Illegal register address in MVB. DatRef and BitRef are included. Command code is of type STRING. Occurs during building an Data Set Package requested from APT or cyclic sending.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

Addr = H'EB30 (=60208)
Illegal register address in MVB. DatRef and BitRef are included. Command code is of type INT. Occurs during building an Data Set Package requested from APT or cyclic sending.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

Addr = H'EB31 (=60209)
Illegal register address in MVB. DatRef and BitRef are included. Command code is
of type INTL, REAL or MIXL. Occurs during building an Data Set Package requested from APT or cyclic sending.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

code	meaning
23 (cont.)	Database error

Addr = H'EB32 (=60210)
Illegal register address in MVB. DatRef and BitRef are included. Command code is of type BOOL. Occurs during building an Data Set Package requested from APT or cyclic sending.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

Addr = H'EB33 (=60211)
Illegal register address in MVB. DatRef and BitRef are included. Command code is of type STRING. Occurs during building an Data Set Package requested from APT or cyclic sending.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

Addr = H'EB34 (=60212)
An attempt to build a DSP with zero registers. This is not allowed. The slave is not able to handle signals with zero registers. MVB is marked with error. Occurs during building an Data Set Package requested from APT or cyclic sending.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

Addr = H'EB35 (=60213, displayed only if DS Log-flag is on)
Illegal Source bit in found MVB, according to the Command Code in the received signal.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = H'XXXXYYYY
XXXX = inCmdCode
YYYY = MVB number

Addr = H'EB3C (=60220)
Illegal scan factor for receive type. MVB has been blocked.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

code	meaning
23 (cont.)	Database error

Addr = H'EB51 (=60241)
Amount of data references in received Data Set Package is too big.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = H'XXXXYYYY
XXXX = Data Set Package size (in DRs)
YYYY = MVB number

Addr = H'EBB5 (=60341)
A 'Setup Request' signal has been received. Node Information DB element (MVINODE) is missing.

Data1 = 3 (CzcNoHit - structure not found)

Data2 = Matching network number

Addr = H'EBB6 (=60342)
A 'Setup Request' signal has been received. Line Characteristics information is missing (MVICHAN).

Data1 = 3 (CzcNoHit - structure not found)

Data2 = Matching network number

Addr = H'EBBE (=60350)
A 'Node Status Information' signal has been received. Appropriate MVINODE to update is missing.

Data1 = 3 (CzcNoHit - structure not found)

Data2 = H'0000XXYY
XX = Matching network number
YY = Matching node number

Addr = H'EBF0 (=60400)
Addressed MVB not found. Occurs during APT request.

Data1 = 3 (CzcNoHit - structure not found)

Data2 = H'XXYYZZ00
XX = Network number
YY = Node number
ZZ = MVB Ident number

code	meaning
23 (cont.)	Database error

Addr = H'EBFF (=60415)
Addressed MVB not found. Occurs during Up Stream Command.

Data1 = 3 (CzcNoHit - structure not found)

Data2 = H'XXYYZZZZ
XX = Network number
YY = Node number
ZZZZ = Lowest two bytes in regAddr

Addr = H'EC09 (=60425)
MVB data reference does not exist. Applies to non-enabled MVBs too.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = H'XXXXYYYY
XXXX = DR instance number
YYYY = MVB instance number

Addr = H'ECB9 (=60601)
Destination in MVB currently unreachable.

Data1 = 4 (CzcBadStruc - bad or inconsistent structure)

Data2 = MVB number

26 **Context in signal error**

Addr = H'EACE (=60110, displayed only if DS Log-flag is on)
A 'Down Stream Response' signal has been received. This signal does not correspond to an earlier sent 'Down Stream Command'.

Data1 = 5 (CzcResponse - error in response signal)

Data2 = H'XXYYZZZZ
XX = Network number
YY = Node number
ZZZZ = Command Code

code	meaning
26 (cont.)	Context in signal error

Addr = H'EAD0 (=60112)
A 'Down Stream Response' has been received. The Response Code is not okey. Error is set on MVB.

Data1 = 5 (CzcResponse - error in response signal)

Data2 = H'XXXXYYYY
XXXX = MVB number
YYYY = Response Code

Addr = H'EAD8 (=60120)
A 'Local Command Response' signal has been received. The Response Code is not okey. Error is set on MVB.

Data1 = 6 (CzcCmdCode - error in command code)

Data2 = H'XXYYZZZZ
XX = Network number
YY = Node number
ZZZZ = Response Code

Addr = H'EAD9 (=60121)
A 'Local Command Response' signal has been received. The Response Code is okey, but the Local Command Code is unknown. Error is set on MVB.

Data1 = 5 (CzcResponse - error in response signal)

Data2 = H'XXYYZZZZ
XX = Network number
YY = Node number
ZZZZ = Response Code

Addr = H'EADA (=60122)
This task instance is waiting for a Local Command Response message, but not the received one

Data1 = 5 (CzcResponse - error in response signal)

Data2 = Command Code from received signal

code	meaning
26 (cont.)	Context in signal error

Addr = H'EADB (=60123)
A 'Local Command Response' signal has been received, but this task instance is not waiting for a Local Command Response.

Data1 = 5 (CzcResponse - error in response signal)

Data2 = Command Code from received signal

Addr = H'EAF7 (=60151, displayed only if DS Log-flag is on)
An 'Up Stream Command' signal has been received. The Command Code in receiving signal can not be identified.

Data1 = 6 (CzcCmdCode - error in command code)

Data2 = H'XXYYZZZZ
XX = Network number
YY = Node number
ZZZZ = Command Code

Addr = H'EB29 (=60201, displayed only if DS Log-flag is on)
Unknown Commad Code in received 'Up Stream Command' signal.

Data1 = 6 (CzcCmdCode - error in command code)

Data2 = H'XXYYZZZZ
XX = Network number
YY = Node number
ZZZZ = Command Code

Addr = H'EB2C (=60204, displayed only if DS Log-flag is on)
Illegal register address in received 'Up Stream Command' signal. Command code is of type INT.

Data1 = 8 (CzcRegister - error in register address)

Data2 = register address

Addr = H'EB2D (=60205, displayed only if DS Log-flag is on)
Illegal register address in received 'Up Stream Command' signal. Command code is of type INTL, REAL or MIXL.

Data1 = 8 (CzcRegister - error in register address)

Data2 = register address

code	meaning
26 (cont.)	Context in signal error

Addr = H'EB2E (=60206, displayed only if DS Log-flag is on)
Illegal register address in received 'Up Stream Command' signal. Command code is of type BOOL.

Data1 = 8 (CzcRegister - error in register address)

Data2 = register address

Addr = H'EB2F (=60207, displayed only if DS Log-flag is on)
Illegal register address in received 'Up Stream Command' signal. Command code is of type STRING.

Data1 = 8 (CzcRegister - error in register address)

Data2 = register address

Addr = H'EB50 (=60240)
Illegal Command Code in receiving Data Set Package.

Data1 = 6 (CzcCmdCode - error in command code)

Data2 = H'XXYYZZZZ
XX = Network number
YY = Node number
ZZZZ = Command Code

Addr = H'EB52 (=60242)
Illegal register address in received signal. Command code is of type INT.

Data1 = 8 (CzcRegister - error in register address)

Data2 = register address

Addr = H'EB53 (=60243)
Illegal register address in received signal. Command code is of type INTL, REAL or MIXL.

Data1 = 8 (CzcRegister - error in register address)

Data2 = register address

code	meaning
26 (cont.)	Context in signal error

Addr = H'EB54 (=60244)
Illegal register address in received signal. Command code is of type BOOL.

Data1 = 8 (CzcRegister - error in register address)

Data2 = register address

Addr = H'EB55 (=60245)
Illegal register address in received signal. Command code is of type STRING.

Data1 = 8 (CzcRegister - error in register address)

Data2 = register address

Addr = H'EB8C (=60300)
Watchdog timeout has expired. Error is set on MVICHAN.

Data1 = 7 (CzcWatchdog)

Data2 = H'XXXXYYYY
XXXX = Slave number (7 - 11)
YYYY = Port number (1 - 2)

Addr = H'EB96 (=60310)
Error sending Node Connect/Disconnect Response Message.

Data1 = 5 (CzcResponse - error in response signal)

Data2 = H'XXYYZZWW
XX = network number
YY = node number
ZZ = 0
WW = 5 - node connection
6 - node disconnection

Addr = H'EBC8 (=60360)
Watchdog message received, but watchdog supervision not activated.

Data1 = 7 (CzcWatchdog - unexpected watchdog message)

Data2 = -

3.4 Console communication (mtype = 18)

System message sent by the console communication sub-system use mtype 18.
The following error codes are used in the “code” field:

code	meaning
0	Target access error. Reason: error in command processor; please report to ABB Industrial Systems AB. Addr = location of error (absolute address) Data1 = internal error code (decimal representation): 0 = instruction buffer overflow 1 = return buffer overflow 2 = no of bytes out of limits in CTREAD or CTCOPY 3 = more bytes from return buffer used than actually was returned 4 = no target connected 5 = illegal target Data2 = -
1	Break handling error. Reason: an illegal call to CXCCSTAT, CXCCCLB or CXCCCIB. The “task” field specifies the illegal caller (if not a user task, please report to ABB Industrial Systems AB). Addr = address to SYSMESS call Data1 = pointer to calling task’s TCB Data2 = -
2	Mass memory utility error. Reason: see data1. Addr = - Data1 = error code (decimal representation): 1 = internal error (please report to ABB Industrial Systems AB). 2 = reset failed. Reason: hardware error 3 = track format failed. Reason: bad disk or hardware error 4 = internal error (please report to ABB Industrial Systems AB). 5 = verify failed. Reason: bad disk 6 = reset failed. Reason: hardware error 32..37= compress failed. Reason: bad disk 3101..= compress failed. Reason: bad disk 3201..= compress failed. Reason: bad disk Data2 = -

code	meaning
3	Symbol handling error. This message is sent by a program that has received an error code from the symbol handler. Reason: internal error, see data1 (please report to ABB Industrial Systems AB). Addr = address in program that received error message from symbol handling Data1 = error code (hexadecimal representation): 1 = illegal operation 2 = one or more symbols were not recognized 4 = wrong type of operation 8 = item or directory already exists 10 = references to subordinated items or directories exists 20 = directory -1 is full 40 = map is full 80 = current directory is full 100 = map or directory is too small to accommodate the new item 200 = syntax error in given symbol 400 = fatal error in some data structure 800 = a no valid map found Please note that data1 is a set of error codes and therefore can consist of more than one of the above listed reasons.
4	Console communication input/output error: some error occurred when Basic Input/output was called from Console communication. Reason: hardware error, or system generation error in parameters concerning device and/or controller. Addr = address in console communication input/output routine Data1 = IQB status word (hexadecimal representation) Data2 = -
5	Console communication message error: a CC message was received but no signal was available. Reason: internal error (please report to ABB Industrial Systems AB) Addr = address to SYSMESS call Data1 = - Data2 = -
6	Console communication heap area overflow Reason: error in command processor (please report to ABB Industrial Systems AB) Addr = address to SYSMESS call Data1 = - Data2 = -

code	meaning
7	Log data for not active log was found in log buffer. Reason: internal error (please report to ABB Industrial Systems AB) Addr = address to SYSMESS call Data1 = log type (decimal representation): 0 = catch log 1 = DB log 2 = signal log Data2 = -
8	Off-line accessor error. Reason: see data1. Error codes 0..5 indicate errors in the target system, while error codes 6..10 indicate internal errors in the service aid (please report to ABB Industrial Systems AB) Addr = address to SYSMESS call Data1 = error code (decimal representation): 0 = target system status error 1 = target system input/output error 2 = target system bus error 3 = target system time-out 4 = target system memory error 5 = target system ADDVER error 6 = internal error 7 = illegal instruction type 8 = illegal address type 9 = return buffer overflow 10 = illegal message type Data2 = -
9	Illegal bit rate specified for target system communication. Reason: internal error (please report to ABB Industrial Systems AB) Addr = address to SYSMESS call Data1 = - Data2 = -
10	Target system utility error. Reason: internal error (please report to ABB Industrial Systems AB) Addr = address to SYSMESS call Data1 = error code (decimal representation): 0 = timer module error Data2 = -

code	meaning
11	Error reported from service aid communication handler. These errors are detected and reported in the service aid. Reason: see data1, please report internal errors to ABB Industrial Systems AB. Addr = address to SYSMESS call Data1 = error code (decimal representation) 0 = message error: the connection is broken 1 = error in port number: internal error 2 = illegal message type: internal error 3 = no answer received: the connection is broken 4 = time out on ENQ: the connection is broken 5 = impossible to send message: the connection is broken 6 = controller error: hardware error Data2 = - For data1 = 10-17 the following applies: Addr = as above, but relative within the task. Data1 = see below Data2 = internal error code (hex.repr.,data1 dependent). Data is divided into four parts (each one byte): sys_err0 (MSB, not used) sys_err1 sys_err2 sys_err3 (LSB) Data1 = error code (decimal representation) 10 = failed to send message Data2 sys_err1 = 1 => slow communication used sys_err3 = 1 => connection is broken, no answer received on ENQ sys_err3 = 2 => connection is broken, no answer received on message sys_err1 = 0 => fast communication used sys_err2 = last response from 68120 sys_err3 = last termination reason from 68120 11 = failed to receive message Data2 sys_err1 = 1 => slow communication used sys_err3 = 1 => ENQ received but no mess. sys_err3 = 2 => only a part of a message was received sys_err3 = 3 => error in termination of message sys_err3 = 4 => no char received after interrupt sys_err1 = 0 => fast communication used sys_err2 = last response from 68120 sys_err3 = last termination reason from 68120

code	meaning
11 (cont.)	Error reported from service aid communication handler. Data1 = error code (decimal representation) 12 = illegal message type, internal error Data2 sys_err = message-number 13 = unexpected message, internal error Data2 sys_err = message-number 14 = controller error, hardware error Data2 - 15 = no channel, internal error Data2 - 16 = illegal port-number, internal error Data2 sys_err = port-number 17 = message already sent to target system but no answer received
12	Error reported from target system communication handler. These errors are detected and reported in the target system. The system messages are printed locally in the target system, or via the service aid. Reason: see data1 (please report internal errors to ABB Industrial Systems AB). Addr = address to SYSMESS call Data1 = error code (decimal representation) 0 = message error: the connection is broken 1 = error in port number: internal error 2 = illegal message type: internal error 3 = no answer received: the connection is broken 4 = time out on ENQ: the connection is broken 5 = impossible to send message: the connection is broken 6 = controller error: hardware error Data2 = - For data1 = 10-17 the following applies: Addr = as above, but relative within the task. Data1 = see below. Data2 = internal error code (hex.repr.,data1 dependent). Data is divided into four parts (each one byte): sys_err0 (MSB, not used) sys_err1 sys_err2 sys_err3 (LSB)

code	meaning
12 (cont.)	Error reported from target system communication handler. Data1 = error code (decimal representation) 10 = failed to send message Data2 sys_err1 = 1 => slow communication used sys_err3 = 1 => connection is broken, no answer received on ENQ sys_err3 = 2 => connection is broken, no answer received on message sys_err1 = 0 => fast communication used sys_err2 = last response from 68120 sys_err3 = last termination reason from 68120 11 = failed to receive message Data2 sys_err1 = 1 => slow communication used sys_err3 = 1 => ENQ received but no mess. sys_err3 = 2 => only a part of a message was received sys_err3 = 3 => error in termination of message sys_err3 = 4 => no char received after interrupt sys_err1 = 0 => fast communication used sys_err2 = last response from 68120 sys_err3 = last termination reason from 68120 12 = illegal message type, internal error Data2 sys_err = message-number 13 = unexpected message, internal error Data2 sys_err = message-number 14 = controller error, hardware error Data2 - 15 = no channel, internal error Data2 - 16 = illegal port-number, internal error Data2 sys_err = port-number 17 = message already sent to target system but no answer received
13	Console communication jump table error. Reason: internal error (please report to ABB Industrial Systems AB) Addr = address to SYSMESS call Data1 = first four characters of called routine Data2 = last four characters of called routine
15	Error in external procedure call: the procedure was not found in the table. Reason: a command processor calls a routine in another command processor that is not present in the system, i.e. the combination of command processors is not consistent. Addr = address to SYSMESS call Data1 = - Data2 = -

code	meaning
16	Erroneous call of 68120 controller. Reason: the system tries to use the controller for fast communication in an erroneous way. Addr = address to SYSMESS call Data1 = - Data2 = -
17	Error during command processor directory init. Reason: error in system generation. Addr = address to SYSMESS call Data1 = error code from symbol handling Data2 = directory number
18	Target system accessor error. Reason: internal error (please report to ABB Industrial Systems AB) Addr = address to SYSMESS call Data1 = error code Data2 = -

3.5 Basic I/O System errors (mtype = 20)

System messages sent by the Basic I/O subsystem use message type 20.
Three groups of error codes are used:

codes	1-10	Device errors
	11-20	Controller errors
	21-	Other errors

code — meaning

1 — Device (e.g. printer) not ready
Reason: no cable connected, printer not switched on, XON signal not generated from printer etc.
Addr = address in the handler
Data1 = Task CXBH26:
Left part of the field: channel number
Right part of the field: slave number and device number are the two characters to the right
Otherwise:
Left part of the field: I/O address
Right part of the field: controller type and device number are the two characters to the right
Data2 = Device status word

2 — Device initiated to off-line mode
Addr = address in the handler
Data1 = See Data1 above
Data2 = Device status word

11 — Controller init error.
Addr = address in the handler
Data1 = See Data1 above
Data2 = Device status word

21 — Bus error (probably when the i/o board was accessed)
Addr = address in the handler
Data1 = See Data1 above
Data2 = Device status word

22 — Data base error.
Addr = address in the handler
Data1 = See Data1 above
Data2 = Device status word

3.6 Operating System Kernel Utility (mtype = 29)

Operating System Kernel Utility functions generates system messages with mtype = 29. The different functions and meanings are listed below:

code	meaning
1	System message from primary task in a redundant system. Address = Message number: 1 = Takeover done Data1 = Backup slot number in the rack Data2 = Geographical address 3 = Backup stopped Data1 = Internal machine state for primary when backup halted Data2 = Backup death cause 5 = RCU hardware error Data1 = Status from RCU driver function Data2 = RCU instance pointer 6 = Task test error Data1 = RCU test number (FB+ Arbitration & CSR test) Data2 = Constant during test 7 = Task error Data1 = 1: Error sending MDT to backup Data2 = Machine state 2: Error sending external state to backup Data2 = (state<<24) + status 8 = Event report Data1 = 0: This is the sole system message using Event Report, but is an error. The state machine has not found any events and this is an error Data2 = 0 9 = Initialization error Data1 = Status of the operation Data2 = 0 10 = RCU test error Data1 = Number of the RCU test Data2 = Internal RCU test status from the test 11 = Backup gave no equality-acknowledge Data1 = Backup external state Data2 = 0 12 = Error updating CTC Data1 = Address to backup CSR space for GPCC Data2 = 0 13 = Error reading CTC Data1 = Error status from ctcDevCtl function Data2 = Error status from ctcGetStatus operation 14 = Error counters Data1 = Maximum accumulated errors Data2 = RCU error register image 15 = FB+ Arbitration error Data1 = 0x0bb10000 + identity of link record Data2 = Status of operation 16 = Backup will not be restarted Data1 = Backup death cause Data2 = Number of backup halts due to this reason

code	meaning
2	Address = 0 - 3: The memory supervision task reports status of a memory board, using Error Detection and Correction Code (EDAC). MasterBase SW*3.0/0 -- MasterBase SW*3.7/20: Address = Error type 0 = Memory board OK. The message is sent after a failing board is replaced with a board without error. Data1 = CPU position Data2 = - 1 = The number of one-bit errors since last check have exceeded a certain limit. Data1 = CPU position Data2 = The start address of the memory board. 2 = The accumulated number of one-bit errors have exceeded a certain limit. Data1 = CPU position Data2 = The start address of the memory board. 3 = Memory board missing. Data1 = CPU position Data2 = The start address of the memory board. MasterBase SW*3.7/21 and later: Address = Error type 0 = Remaining one-bit error(s) has been “cleared” by a repeating re-write of the memory. Data1 = CPU position Data2 = The start address of the memory board. 1 = One or more one-bit error(s) has been detected and corrected, i.e. the memory is successfully updated with the corrected value. Data1 = CPU position Data2 = The start address of the memory board. 2 = One or more remaining one-bit error(s) has been detected, i.e. it was not possible to update the memory with the corrected value. Note that the EDAC circuit still is able to correct the error when reading the memory. Data1 = CPU position Data2 = The start address of the memory board. 3 = Memory board missing. Data1 = CPU position Data2 = The start address of the memory board.

code	meaning
2	Address = H'10 - H'16: The PCMCIA card supervision function reports status of a PCMCIA card. Address = Error type H'10 = A removed card is replaced with an identical one. Data1=slot number Data2=sub module number H'11 = A card checksum error is detected. Data1=slot number Data2=sub module number H'12 = A card is removed. Data1=slot number Data2=sub module number H'13 = A removed card is replaced with a card, which differ from the original one. Data1=slot number Data2=sub module number H'14 = A card semaphore is busy too long time. Data1=slot number Data2=sub module number H'15 = A card semaphore is free again. Data1=slot number Data2=sub module number H'16 = A FFS2 card is replaced with another FFS2 card. Note that the new card is accepted and the device will resume the checksum checking but a system message is sent to inform that the replaced FFS2 card differs from the previous one. Data1=slot number Data2=sub module number
3	Reserved.

code	meaning
4	This code is used in the time synchronization task. Address = Error type 1 = More than 3 consecutive sync pulses are lost. Earlier calculated adjust interval is used for continuous time adjustment until the sync pulses occurs again. Data1 = - Data2 = - 2 = The calculated time adjustment (to keep the system time synchronized with the external clock) becomes too big, i.e. the adjust interval too small. The external time synchronization function is set off-line and is automatically restarted (set on-line) when the sync pulses returns at regular intervals with a difference from the system time less than 1 second per sync interval. Data1 = Difference per minute (in units of 0.1 milliseconds) Data2 = -
5	Reserved.
6	Reserved.
7	System message from Ux channel services. Address = Error type 0 = Error from cUxCreate Data1 = Channel ID Data2 = Returned error code from cUxCreate 1 = Error from cUxSend Data1 = Channel ID Data2 = Returned error code from cUxSend 2 = Error from vhiDevCtl (CmxFilledFifoElGet) Data1 = Channel ID Data2 = Returned error code form vhiDevCtl. 3 = Error from vhiRData Data1 = Channel ID Data2 = Returned error code from vhiRData 4 = Out of sync Data1 = Channel ID Data2 = Magic no 5 = Read buffer too small Data1 = Channel ID Data2 = Signal size 6 = Unexpected task exception Data1 = Channel ID Data2 = Exception code

code	meaning
8	Report from CX Start-up function. Address = Error type. 1 = Start-up semaphore not released within specified time. Data1 = TCB-address of semaphore holding task. Data2 = - 2 = Init of start-up semaphore failed. Data1 = - Data1 = - 3 = Start-up semaphore not defined. Data1 = - Data2 = -
9	System message channel overflow (UNIX system message). Addr = 0 Data1 = Channel ID Data2 = -
10	Report from Virtual Hardware Interface in UXBase (VHI_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
11	Report from PU510 controller in UXBase (DXDSB_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
12	Report from PU510 driver in UXBase (DXCSB_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
13	Report from PU510 channel controller in UXBase (DXCCH_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
14	Report from PU510 message exchange channel controller in UXBase (DXCMX_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
15	Report from PU510 message exchange channel driver in UXBase (DXDMX_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -

code	meaning
16	Report from PU510 terminal channel controller in UXBase (DXCTM_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
17	Report from PU510 terminal channel driver in UXBase (DXDTM_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
18	Report from MP510 pseudo driver in UXBase (DXDMP_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
19	Report from Basic Support software in UXBase (BS_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
20	Report from UXBase (DXUER_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
21	Report from PU510 time synchronization process in UXBase (DXUTS_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
22	Report from Pseudo HIL driver in UXBase (DXPHL_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
23	Report from PU510 printer handler in UXBase (DXUPR_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
24	Report from PU510 process supervisor in UXbase (DXUSV_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
25	Report from UXbase software for HP-UX UNIX system (DXHPUX_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -

code	meaning
26	Report from CX file server in UXBase (DXUMM_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
27	Report from UXBase (DXUBO_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
28	Report from PU510 terminal emulator in UXBase (DXUTM_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
29	Report from PU510 trace driver in UXBase (DXDTR_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
30	Report from UXBase (DXUTO_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
31	Report from file format software in UXBase (DXUFF_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
32	Report from UNIX console handler (DXUCH_COMP). Addr = 3 A UNIX console message has been generated. Could be serious since these kind of messages are rare. Examine the console message in the UNIX system message log. (DXUCH_UX_CONS_MSG). Data1 = - Data2 = -
33	Report from UXBASE system status (DXUSS_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
34	Report from UXBase support software, dxPU510upprt (DXUSP_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -

code	meaning
35	Report from PU510 Boot/Dump software (DXUBD_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
36	Report from UXBase (DXUVI_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
37	Report from PU510 configuration software, e.g. ECONFIG and DCONFIG (DXUCF_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
38	Report from UXBase (DXUDU_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
39	Report from UXBase (DXUSU_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
40	Report from PU510 stall supervisor in UXBase (DXUST_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
41	Report from PU510 Command channel software in UXBase (DXUCC_COMP). Addr = The UNIX system message number. Data1 = - Data2 = -
42 to 49	Reserved for future UXBase components.
50	Report from VxBase “Illegal component” Addr = VxBase system message number Data1 = - Data2 = -
51	Report from VxBase “Date and Time component”. Addr = VxBase system message number Data1 = - Data2 = -
52	Report from VxBase “Assertion component”. Addr = VxBase system message number Data1 = - Data2 = -

code	meaning
53	Report from VxBase “Stall Supervision component”. Addr = VxBase system message number Data1 = Priority of stalled task Data2 = Identity of stalled task
54 to 69	Reserved for VxBase component.
70	System message from backup task in a redundant system. Address = Message number: 6 = Test error Data1 = Status from FB+ interface function Data2 = (100*primary slot no) + message number 7 = Task error Data1 = 1: CPU instruction test failure Data2 = 0 2: Failure to reply to primary Data2 = Status 3: Failure to update CTC Data2 = Status 9 = Checksum error Data1 = 1: Error checksumming checksum table Data2 = Number of bad records 2: Checksum head differs from primary table Data2 = MTD reference 3: Checksum differ from primary tables Data2 = Checksum error location (1k block) 10 = Thread stack overwritten Data1 = Index of the “stack” safety record where the overwriting was found Data2 = Garbage 11 = Thread executive error Data1 = 1: No current thread instance (no task) Data2 = 0 2: No more free threads at create Data2 = 0 3: Thread instance not found Data2 = ID of thread not found 12 = Memory error Data1 = Status from memory object Data2 = 0 14 = EDAC warning Data1 = Start address where error found Data2 = Delta errors

code	meaning
71	System message from primary driver part in a redundant system. Address = Message number: 5 = RCU hardware error Data1 = RCU hardware error register content Data2 = Event list 6 = Test error Data1 = Number of dual RCU test (8 or 9) Data2 = (index<<24) + data[index] 9 = GPCC error Data1 = The code of the message Data2 = 0 10 = Bus error Data1 = Event list Data2 = Garbage 11 = Default position reached in switch Data1 = 1: GPCC message code unknown Data2 = Code received 2: GPCC message command unknown Data2 = Command received 3: RCU user interrupt reason unknown Data2 = Interrupt reason
72	Reserved for backup driver part in a redundant system

Chapter 4 CPU Board Lamp Indicators

On the upper part of the front panel of the CPU board there are a number of lamp indicators:

Different CPU boards have different numbers of indicators, see Table 4-1. They indicate different, steady or temporary, error conditions of the system.

Table 4-1. CPU board indicators

Indicator	PM510	PM150	PU520	PU530
F	*	*	*	*
RUN	*	*	*	*
HLT	*	*		*
TO	*	*		*
DUAL	*			
BAT	*	*		
DCOK		*		
INHIB		*		
MIB		*		
Seven segment displays	*	*		

4.1 F

This indication is asserted by application software at detected fault.

4.2 RUN

Indicates that the CPU is running. It is turned off if the CPU stops.

4.3 HLT

The HLT indicator is lighted when the processor has halted due to a double bus error condition on the processor bus. (For detailed information of the reason for that see M 68000 User's Manual).

4.4 TO

The TO indicator indicates, when lighted, that a bus error (bus time-out) condition is present on the processor bus. That condition can be either steady or temporary. It can be temporary if the system, in that particular situation, is able to recover from the reason of the bus error. That is the situation for example during cyclic check of missing boards. During the system start-up phase the TO indicator is lighting and is switch off when the start-up is completed. Those temporary lightings of the TO indicator is due to normal operations and are not to be seen as errors. The

TO indicator is steady lighting when the system has halted due to the bus error condition and then the RUN is turned off.

4.5 DUAL

Indicates that a redundant CPU is present and active.

4.6 BAT

Indicates that the backup power supply is connected.

4.7 DCOK

Indicates that at least one of the two possible 24V sources is within specified level.

4.8 INHIB

Indicates that INHIB is active on the PBC bus.

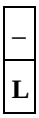
4.9 MIB

The MIB indicator is lighted when PM150 is the current bus master.

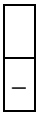
4.10 Seven segment displays

The two seven segment displays are used to indicate different working modes and, at system stop, to indicate a system halt code.

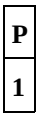
The working modes are:



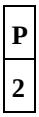
System loading in progress



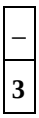
System start-up in progress



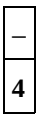
System running in operation mode



System running in configuration mode



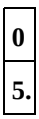
System stopped by operator, i.e. the start mode selector is in STOP position



System running in off-line mode

At system halt a halt code is displayed (see Section 2.3 System halt error code interpretation). The upper indicator gives the most significant digit (hex) and the lower indicator gives the least significant digit (hex). On the lower indicator the decimal point is also lighting.

Example:



Bus time-out error in system or kernel mode

REVISION

Rev. Ind.	Page (P) Chapt (C)	Description	Date Dept./Init
-		First version for MasterBase 3.x	1994-09-12 SEISY/LKSS SOH
A	C 2.3	Halt code H'0006, H'0008, H'0018-H'0022, H'0025-H'002B, H'002D-H'0031, H'0034, H'0035, H'0039 and H'003A marked as reserved. Description of halt code H'0007, H'000B and H'0047 corrected Description of halt code H'0049 added	1995-08-09 SEISY/LKSS SOH
	C 3.3.1	Description of system message 17:8:4 and 17:9:1 corrected	
	C 3.6	Description of system message 29:1 corrected Description of system message 29:70, 29:71 and 29:72 added	
B	C 3.3.6	Halt codes and description of system messages are added for MVI-Data Set with MVB API.	1998-04-17 SEISY/LKT AN
	C 3.6	Description of system message 29:2 corrected.	1998-04-17 SEISY/LKT SOH

3BSE000331/B

April 1998